UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/698,762	10/31/2003	Bimal Mehta	MSFT-2748/302029.1	2802

41505 7590 03/15/2010
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
ANYA, CHARLES E	
ART UNIT	PAPER NUMBER
2194	
MAIL DATE	DELIVERY MODE
03/15/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/698,762	MEHTA ET AL.	
	Examiner	Art Unit	
	CHARLES E. ANYA	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3/* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 December 2009*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3 and 5-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3 and 5-34* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-3 and 5-34 are pending in this application.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1, 2, 5-10, 12-24 and 26-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 2003/0093500 A1 to Khodabakchian et al. in view of U.S. Pat. No. 7,036,045 B2 issued to Broussard et al.

3. As to claim 1, Khodabakchian teaches an asynchronous messaging architecture for processing messages, comprising:

a processor operatively coupled to a computer readable storage medium including computer executable instruction (figure 8 page 6 paragraphs 0070-0076) for:

executing an instance of an automated business process ("…process…" page 1 paragraph 0017, page 3 paragraph 0035), the automated business process including response processing code including exception handling code specifying error compensation ("…compensation rules and exception handlers for a specific scenario…" page 2 paragraphs 0024/0025);

executing a program manager configured to manage the instance of the automated business process (Web Service Orchestration Server 102);

the program manager further configured to detect when the instance of the automated business process is waiting for a response to a message ("…suspended until a response…" page 1 paragraph 0017, figure 5 page 4 paragraph 0041), wherein a response indicates a success or failure of the message ("…suspended until a response…" page 1 paragraph 0017, page 3 paragraph 0035, Block 706 page 4 paragraph 0047);

the program manager further configured to store, when the instance is waiting for the response, at least a part of state information associated with the instance in a database and remove the instance from active memory ("...Passivation..." page 3 paragraph 0035, Block 510 page 4 paragraph 0041, "…passivates the states of the process…" page 4 paragraph 0041);

the program manager further configured to determine when the response associated with the instance has been received ("…When the response…is received…" page 3 paragraph 0035, Block 508 page 4 paragraph 0041) and the program manager further configured to restore the instance from the database into memory and pass the instance the message ("…the process is reactivated…" page 3 paragraph 0035, Block 514 page 4 paragraph 0041); and

the instance further configured to process the response using response processing code for the instance ("…Execution of the process…" page 3 paragraph

0035, Block 516 page 4 paragraph 0041) and handling asynchronous messaging errors synchronously for the instance (Debugger 204 page 2 paragraphs 0026/0027).

Khodabakchian is silent with reference to the instance further configured to process the response using the exception handling code within the instance.

Broussard teaches the instance further configured to process the response using the exception handling code within the instance (figure 4 "…**Within** application logic 50 is a piece of code containing a "try-catch" block 56 to deal with any "Exception e" errors that might occur while "Do some business logic" is executed…" Col. 4 Ln. 52 – 64).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Khodabakchian with the teaching of Broussard because the teaching of Broussard would improve the system of Khodabachian by providing a methodical process (try-catch block) for signaling and handling usual conditions including errors or defects, in a computer program by testing the block of code for errors and catching/executing the block of code if an error occurs, thus, reducing the number of errors or defects in the computer program.

4. As to claim 2, Broussard teaches the architecture of claim 1, wherein the response processing code is a try-catch block ("…"try-catch" block 56…" Col. 4 Ln. 52 – 64).

5. As to claim 5, Khodabachian teaches the architecture of claim 1, wherein the response is received on a port defined by the instance (figure 5 page 4 paragraph 0041).

6. As to claim 6, Khodabakchian teaches the architecture of claim 1, wherein the response is a response indicative of whether or not the message was received by an intended recipient (figure 5 page 4 paragraph 0041).

7. As to claim 7, Khodabakchian teaches a method for processing a message in an asynchronous architecture, comprising:

determining that a response to a message sent by an instance (process) of software code is to be received (Block 508 page 4 paragraph 0041), wherein the response indicates a success or failure of the message (Block 706 page 4 paragraph 0047);

determining whether the response has been received and, if the response has not been received, storing the instance of the software code in memory, thereby suspending the instance ("...Passivation..." page 3 paragraph 0035, Block 508 page 4 paragraph 0041, Block 510 page 4 paragraph 0041, "…passivates the states of the process…" page 4 paragraph 0041);

receiving the response ("…When the response…is received…" page 3 paragraph 0035, Block 512 page 4 paragraph 0041) and resuming the instance ("…Execution of the process…" page 3 paragraph 0035, Block 514 page 4 paragraph 0041); and processing the response using response processing code within the instance according to the success or failure of the message ("…Execution of the process…" page 3 paragraph 0035, Block 516 page 4 paragraph 0041), wherein a response processing code having failure handling functionality specifying error compensation ("…compensation rules and exception handlers for a specific scenario…" page 2 paragraphs 0024/0025) and handling asynchronous messaging errors synchronously for the instance (Debugger 204 page 2 paragraphs 0026/0027).

Khodabakchian does not explicitly teach a response processing code within the instance having failure handling functionality.

Broussard teaches a response processing code within the instance having failure handling functionality (figure 4 "…**Within** application logic 50 is a piece of code containing a "try-catch" block 56 to deal with any "Exception e" errors that might occur while "Do some business logic" is executed…" Col. 4 Ln. 52 – 64).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Khodabakchian with the teaching of Broussard because the teaching of Broussard would improve the system of Khodabachian by providing a methodical process (try-catch block) for signaling and handling usual conditions including errors or defects, in a computer program by testing the block of code for errors and catching/executing the block of code if an error occurs, thus, reducing the number of errors or defects in the computer program.

8. As to claims 8, 22 and 32, see the rejection of claim 2 above.

9. As to claim 9, Broussard teaches the method of claim 8, wherein processing the response comprises determining whether the response indicates a failure and, if so, processing the response using the catch block ("…"try-catch" block 56…" Col. 4 Ln. 52 – 64).

10. As to claim 10, Khodabakchian teaches the method of claim 9, further comprising, if the response indicates a success, processing the response by way of the instance of the software code (Block 516 page 4 paragraph 0041).

11. As to claim 12, Khodabakchian teaches the method of claim 7, wherein storing the instance comprises storing the instance in a database and removing the instance from active memory ("…passivates the states of the process…using the stored data associated with the process…" page 3 paragraph 0035, page 4 paragraph 0041, page 6 paragraph 0068).

12. As to claim 13, Khodabakchian teaches the method of claim 12, wherein resuming the instance comprises removing the instance from the database and restoring the instance to active memory ("…passivates the states of the process…using the stored data associated with the process…" page 4 paragraph 0041, page 6 paragraph 0068).

13. As to claims 14 and 34, see the rejection of claim 5 above.

14. As to claim 15, Khodabachian teaches the method of claim 7, wherein the asynchronous architecture is implemented by way of distributed business process automation software (Web Services 104 page 2 paragraph 0020).

15. As to claim 16, Khodabachian teaches the method of claim 7, wherein the message is to be received by a remote computer (page 3 paragraph 0034).

16. As to claim 17, see the rejection of claims 1 and 2 above.

17. As to claims 18 and 23, see the rejection of claim 9 above.

18. As to claim 19, Khodabakchian teaches the method of claim 18, further comprising, if the response is indicative of a success, processing the response within the instance of the automation software and logically after the catch block (Block 516 page 4 paragraph 0041).

19. As to claim 20, see the rejection of claim 14 above.

20. As to claims 21 and 31, see the rejection of claims 1 and 7 respectively.

21. As to claims 24 and 33, see the rejection of claim 19 above.

22. As to claims 26-30, see the rejection of claims 12-16 respectively.

23. As to claim 34, see the rejection of claim 14 above.

24. Claims 3, 11 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 2003/0093500 A1 to Khodabakchian et al. in view of U.S. Pat. No. 7,036,045 B2 issued to Broussard et al. as applied to claims 1, 7 and 21 above, and further in view of U.S. Pub. No. 2003/0204835 A1 to Budhiraja et al.

25. As to claim 3, Broussard and Khodabakchian are silent with reference to the architecture of claim 1, wherein storing the instance takes place after a predetermined time.

Budhiraja teaches the architecture of claim 1, wherein storing the instance takes place after a predetermined time ("…checkpoint…" page 3 paragraphs 0037/0041/0048).

It would have been to one of ordinary skill in the art at the time the invention was made to modify the system of Broussard and Khodabakchian with the teaching of Budhiraja because the teaching of Budhiraja would improve the system of Broussard and Khodabakchian by providing a technique for inserting fault tolerance into computing systems which includes storing a snapshot of a current application state, and using it for restarting execution in case of failure.

26. As to claims 11 and 25, see the rejection of claim 3 above.

Response to Arguments

Applicant's arguments filed 12/22/09 have been fully considered but they are not persuasive.

Applicant argues in substance that (1) the Broussard prior art teaches away from the claimed invention because it's try-catch block 56 is not implemented within it's application logic, (2), the Broussard prior art does not perform an error handling as claimed invention requires, (3) there is no legitimate motivation for combining the Khodabakchian and Broussard prior arts and (4) the Budhiraja prior art does not teach the step of storing the instance information after a predetermined time.

After comprehensive consideration of the prior arts the Examiner respectfully traverses Applicant's arguments.

As to point (1), the Broussard prior art discloses a system and method for catching and dumping all or some exceptions in an object-oriented environment. The catching and dumping of exceptions occur during the running of and application using ""try-catch" block 56" (Col. 2 Ln. 1 – 10, Col. 4 Ln. 57 – 60). As the current rejection indicates the Broussard prior art discloses that the ""try-catch" block 56" is provided **within the application** (Col. 4 Ln. 57 – 60) and this passage is functionally equivalent to claimed "…handle exceptions using the exception handling code **within the instance**…".

As to point (2), the Broussard prior art discloses that when an application is run, and an error/exception occurs, a checking system 22 will identify exception object classes that match the defined exception subset. The checking system 22 can be implemented as a software routine that compares the object class of an exception with the object classes defined in the exception subset. If a match is found, then the information associated with the exception is collected from the JVM 18 and outputted by exception dumping system 24 to log file 34. Exception dumping system 24 may be implemented, for instance, by a software routine that opens log file 34 and writes the exception data to the log file 34 (Col. 4 Ln. 1 – 18).

This process of detecting, catching, matching and dumping of errors or exceptions using the checking and dumping systems is functional equivalent to claimed "…**handle exceptions using the exception handling code** within the instance…"

As to point (3), the current rejection makes this argument moot.

As to point (4), checkpoint-recover is a common technique for equipping a program or system with fault tolerant qualities. It allows systems to recover after some fault in the system that causes a task or process to fail, or be aborted in some way. The basic idea behind checkpoint-recover is the saving and restoration of system or process state. It saves the current state of the process **periodically** or before critical code sections. When a system or process state is checkpointed, this state is saved to non-volatile storage. In the event of a system failure, the internal state of the system or process can be restored, and it can continue service from the point at which its state was last saved. Typically this involves restarting the failed process or system, and providing some parameter indicating that there is state to be recovered.

The saving of snapshot of the process state may be **scheduled periodically** during program execution. Typically this is accomplished by pausing the operation of the process whose state is to be saved, and copying the memory pages into non-volatile storage.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHARLES E. ANYA whose telephone number is (571)272-3757. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Hyung S. Sough/
Supervisory Patent Examiner, Art Unit 2194
03/12/10

cea.